UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/608,757	06/27/2003	Yao Wang	EMC-01-141CIP2	7174

24227 7590 10/14/2008
EMC CORPORATION
OFFICE OF THE GENERAL COUNSEL
176 SOUTH STREET
HOPKINTON, MA 01748

EXAMINER	
ADAMS, CHARLES D	
ART UNIT	PAPER NUMBER
2164	
MAIL DATE	DELIVERY MODE
10/14/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/608,757	**Applicant(s)** WANG ET AL.	
	Examiner CHARLES D. ADAMS	**Art Unit** 2164	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 June 2008*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-17* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-17* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

Remarks

1. In response to communications filed on 27 June 2008, claims 1, 6, 10, and 14 are amended. Claims 1-17 are pending in the application.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 1-9 are rejected under 35 U.S.C. 101 because the claims are not directed towards one of the statutory classes. An architecture is neither a process, a machine, a manufacture, nor a composition of matter. An architecture is a software-based framework for implementing a system. While applicant argues that the architecture comprises agents embodied on physical mediums, it is unclear how a software computer architecture is capable of containing physical mediums. Thus, it is unclear which of the statutory classes claims 1-9 are directed towards.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sicola et al. (US Pre-Grant Publication 2004/0064639) in view of Mashayekhi et al. (US Patent 6,922,791).

As to claim 1, Sicola et al. teaches:

A data transfer server (see paragraphs [0049] and [0054]);

a primary software agent hosted on said data storage system, said primary software agent, in communication with the data transfer server, the primary software agent configured for performing data transfer operations in response to commands from the data transfer server (see paragraphs [0049], “storage array controllers”, [0058], “remote copy set operation”, and [0050], which teaches ‘host’ and ‘disaster-tolerant’ paths. Also see paragraph [0054], which discusses software running on the hosts, that is used for dynamic failover between storage paths);

One or more failover software agents, each failover software agent residing on a host, in communication with the primary software agent, the data transfer server, and at least one of the two data storage systems, the failover software agents being remote from the primary software agent (see paragraphs [0049]-[0051], [0054] and Figure 2. All storage subsystems and all hosts are visible to each other over the SAN. Therefore, as they are connected by various network components, they are remote),

Sicola et al. does not explicitly teach wherein said primary software agent further represents a failover software agent for another of said primary software agents in another one of said data storage systems;

Mashayekhi et al. teaches wherein said primary software agent further represents a failover software agent for another of said primary software agents in another one of said data storage systems (see Mashayekhi et al. 6:25-38 and 1:58-16, and 7:42-51 and Figure 2a. There are multiple ways listed in Mashayekhi et al. which provide for a primary software agent serving as a failover agent for another primary agent);

Sicola et al. as modified teaches:

A failover protocol for determining an order which said software agents, within a communication path of a data transfer, are designated to take over the data transfer operation in response to one or more data transfer commands when a failure of one or more of said software agents is determined, said protocol being determined during configuration of said computer architecture (see Mashaykhi et al. 6:25-38 and 1:58-16 and Sicola et al. paragraph [0054]).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Sicola et al. by the teaching of Mashayekhi et al., since Mashayekhi et al. teaches that “the implementation of a failover policy can be crucial when considering the overall high availability of a cluster” (see 1:28-30).

As to claim 2, Sicola et al. as modified teaches wherein the data transfer operation is a replication of data within the data storage environment (see Sicola et al. paragraph [0058]).

As to claim 3, Sicola et al. as modified teaches wherein server commands to the software agents are sent over a network in accordance with an IP protocol (see Sicola et al. paragraph [0053]).

As to claim 4, Sicola et al. as modified teaches wherein the software agents communicate with the at least one data storage system over the network in accordance with a Fibre Channel protocol (see Sicola et al. paragraph [0052]).

As to claim 5, Sicola et al. as modified teaches wherein a predetermined hierarchal relationship is followed by the data transfer server to select the order in which the failover software agents are commanded to take over the work of the one or more determined failed software agents (see Mashayekhi et al. 6:25-38 and 1:58-16, and 7:42-51).

As to claim 6, Sicola et al. teaches:

A data replication management server (see paragraphs [0049] and [0054]);

A software agent, designated as primary software agent, hosted on said data storage system, said primary software agent, in communication with at least one of the two data storage systems and the data replication management server, the primary software agent configured for performing data replication operations in response to commands from the data replication management server (see paragraphs [0049]-[0050], [0054], and paragraph [0058]);

One or more failover software agents, each failover software agent residing on a host, in communication with the primary software agent, the data replication management server, and at least one of the two data storage systems, the failover software agents being remote from the primary software agent (see paragraphs [0049]-[0051], [0054] and Figure 2)

Sicola et al. does not explicitly teach wherein said primary software agent further represents a failover software agent for another of said primary software agents in another one of said data storage systems;

Mashayekhi et al. teaches wherein said primary software agent further represents a failover software agent for another of said primary software agents in another one of said data storage systems (see Mashayekhi et al. 6:25-38 and 1:58-2:16, and 7:42-51 and Figure 2a);

Sicola et al. as modified teaches a failover protocol for determining an order in which said software agents, within a communication path of a data transfer (see Sicola et al. paragraphs [0049]-[0052] and Figure 2), are designated to take over the data transfer operation in response to one or more data transfer commands when a failure of one or more of said software agents is determined, said protocol being determined during configuration of said computer architecture (see Mashayekhi et al. 6:25-38 and 1:58-16, and 7:42-51 and Sicola et al. paragraph [0054]).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Sicola et al. by the teaching of Mashayekhi et al., since Mashayekhi et al. teaches that “the implementation of a failover policy can be crucial when considering the overall high availability of a cluster” (see 1:28-30).

As to claim 7, Sicola et al. as modified teaches wherein server commands to the software agents are sent over a network in accordance with an IP protocol (see paragraph [0053]. An “Internet Link” would inherently use an “IP protocol”).

As to claim 8, Sicola et al. as modified teaches wherein the software agents communicate with the at least one data storage system over the network in accordance with a Fibre Channel protocol (see paragraph [0052]).

As to claim 9, Sicola et al. as modified teaches wherein the data replication management server uses a predetermined hierarchal relationship to select the order in which designated ones of the failover software agents are commanded to take over the work of the one or more determined failed software agents (see Mashayekhi et al. 6:25-38 and 1:58-16, and 7:42-51).

As to claim 10, Sicola et al. teaches a method for managing fault-tolerant resources for replication of data stored in a data storage environment including at least two data storage systems, and wherein data is replicated (see paragraph [0058]) from one of the at least two data storage systems to at least one other data storage system of the at least two data storage systems (see paragraph [0049]), and at least one software agent in communication with at least one data replication management server for managing the fault tolerant resources (see paragraph [0049]), the method comprising:

configuring one or more software agents as failover agents, each failover software agent residing on a host, that are in remote communication with another software agent, designated as primary software agent, which is also in communication with the data replication management server, and at least one of the two data storage systems (see paragraphs [0049]-[0051], [0054] and [0058])

Sicola et al. does not teach wherein said primary software agent further represents a failover software agent for another of said primary software agents in another one of said data storage systems;

Mashayekhi et al. teaches wherein said primary software agent further represents a failover software agent for another of said primary software agents in another one of said data storage systems (see Mashayekhi et al. 6:25-38 and 1:58-2:16, and 7:42-51 and Figure 2a);

Sicola et al. as modified teaches establishing a failover protocol for determining an order in which said software agents, within a communication path of the data transfer (see Sicola et al. paragraphs [0049]-[0052] and Figure 2), are designated to take over the data transfer operation in response to one or more data transfer commands when a failure of one or more of said software agents is determined, said protocol being determined during configuration of said computer architecture (see Mashayekhi et al. 6:25-38 and 1:58-16, and 7:42-51 and Sicola et al. paragraph [0054]).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Sicola et al. by the teaching of Mashayekhi et al., since Mashayekhi et al. teaches that "the implementation of a failover policy can be crucial when considering the overall high availability of a cluster" (see 1:28-30).

As to claim 11, Sicola et al. as modified teaches wherein server commands to the software agents are sent over a network in accordance with an IP protocol (see paragraph [0053]).

As to claim 12, Sicola et al. teaches wherein the software agents communicate with the at least one data storage system over the network in accordance with a Fibre Channel protocol (see paragraph [0052]).

As to claim 13, Sicola et al. as modified teaches wherein the data replication management server uses a predetermined hierarchal relationship to select the order in which designated ones of the failover software agents is commanded to take over the work of the one or more determined failed software agents (see Mashayekhi et al. 6:25-38 and 1:58-16, and 7:42-51).

As to claim 14, Sicola et al. as modified teaches:

A data replication management server (see paragraphs [0049] and [0054]);

A software agent, designated as primary software agent, hosted on said data storage system, said primary software agent in communication with at least one of the two data storage systems and the data replication management server, the primary software agent configured for performing data replication operations in response to commands from the data replication management server (see paragraphs [0049]-[0050], [0054] and [0058]);

One or more failover software agents, each failover software agent residing on a host, in communication with the primary software agent, the data replication management server, and at least one of the two data storage systems, the failover software agents being remote from the primary software agent (see paragraphs [0049]-[0051], [0054] and Figure 2);

Sicola et al. does not explicitly teach wherein said primary software agent further represents a failover software agent for another of said primary software agents in another one of said data storage systems;

Mashayekhi et al. teaches wherein said primary software agent further represents a failover software agent for another of said primary software agents in another one of said data storage systems (see Mashayekhi et al. 6:25-38 and 1:58-16, and 7:42-51 and Figure 2a);

Sicola et al. as modified teaches a computer-executable program for carrying out a failover protocol for determining an order in which said software agents, within a communication path of the data transfer (see Sicola et al. paragraphs [0049]-[0052] and Figure 2), are designated to take over the data transfer operation in response to one or more data transfer commands when a failure of one or more of said software agents is determined (see Mashayekhi et al. 6:25-38 and 1:58-2:16, and 7:42-51 and Sicola et al. paragraph [0054]).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Sicola et al. by the teaching of Mashayekhi et al., since Mashayekhi et al. teaches that "the implementation of a failover policy can be crucial when considering the overall high availability of a cluster" (see 1:28-30).

As to claim 15, Sicola et al. as modified teaches wherein server commands to the software agents are sent over a network in accordance with an IP protocol (see paragraph [0053]. An "Internet Link" would inherently use an "IP protocol").

As to claim 16, Sicola et al. as modified teaches wherein the software agents communicate with the at least one data storage system over the network in accordance with a Fibre Channel protocol (see paragraph [0052]).

As to claim 17, Sicola et al. as modified teaches wherein the data replication management server uses a predetermined hierarchal relationship to select the order in which designated ones of the failover software agents are commanded to take over the work of the one or more determined failed software agents (see Mashayekhi et al. 6:25-38 and 1:58-16, and 7:42-51).

Response to Arguments

6. Applicant's arguments filed 27 June 2008 have been fully considered but they are not persuasive.

In regards to the rejections under 35 U.S.C. 101, Applicant argues that “the invention as claimed is patentable subject matter as the software executing on the general purpose general computer transforms the general purpose digital computer into a special purpose digital computer" and “claims 1, 6, 10, and 14, are architecture, method, and system versions of the current invention”.

In response to these arguments, it is noted that claims 1 and 6 are directed towards a computer architecture. A computer architecture is a framework of software or hardware components. The architecture itself need not be limited to any physical embodiment, nor, in this case, is the architecture limited in such a manner. It is a blueprint or design document. It is noted that an 'architecture' is neither a process, machine, manufacture, nor composition of matter. As such, it is unclear what statutory class these claims belong to.

Applicant argued that Sicola et al. cannot meet the claimed invention, as Sicola et al. “states his method provides such that ‘no special host software is required to implement the above features because all replication functionality is totally self contained within each array controller and automatically done without user intervention'". In response to that argument, it is noted that Sicola et al. teaches, in paragraph [0054], that " "the hosts typically run multi-pathing software (known in the art) that dynamically allows failover between storage paths as well as static load balancing of storage volumes (LUNs) between paths to the controlle—based storage arrays 201/202 and 211/212. The configuration of system 100 allows for applications using either of the storage arrays 203/213 to continue running given any failure of either fabric 103A/103B or either of the storage arrays". It is noted that Sicola et al. explicitly mentions hosts storing multi-pathing software to accomplish failover functionality.

Applicant argues that Mashayekhi et al. “does not disclose ‘software agents,' a priori, running on hosts, rather Applicants assert it discloses ‘assigning applications’”. In response to this argument, it is noted that Sicola et al. is relied upon to teach the ‘software agents’ of the claims, and Mashayekhi et al. is relied upon to further teach that one system that is regarded as a primary system may be relied upon as a failover system in the case of a failure at another system.

Applicants argue that “no such ‘software agents’ exist in either reference. However, assuming arguendo, that such a reference to ‘software agents’ did occur, there is no reason as to why such they would be located on hosts. Sicola explicitly states his method features requiring ‘no special host software,' rather his functionality is ‘within each array controller.’ Mashayeki is concerned with ‘assigning applications.’ Neither references would enable one skilled in the art to create software agents and place them on hosts, which would directly remove one of Sicola’s features and would be redundant in light of Mashayeki’s ‘assigning applications.'" In response to this argument, it is noted that Sicola et al. explicitly states wherein each host runs multi-pathing software that dynamically allows failover between storage paths as well as static load balancing of storage volumes between the paths to the controller-based storage arrays (see paragraph [0054]). While no ‘special’ host software may be needed to implement the features of Sicola et al. (see paragraph [0019]), it is clear that Sicola et al. states that the hosts each run multi-pathing software (see paragraph [0054]).

Applicants also “assert Mikashayeki determines which node is used to recover the failed node by ‘in the weight determining step, the weight of every one of the surviving nodes is determined,' whereas, the current invention differs in that 'said protocol… [is] determined during configuration of said computer architecture'. Mikashayeki’s determine is ‘following failure of a node.' Performing his determination of a failover node following failure of a node, Mikashayeki does not disclose that this occurs 'during configuration of said computer architecture.'" In response to this argument, Examiner notes that the first cited portions of Mikashayeki et al., 1:58-2:16, simply have a unidirectional linear series for a failover policy, and also states "each node in the cluster may failover only to one single node that has been designated prior to the time of failure" (see 2:11-13). Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have implemented the protocols listed in Mikashayeki et al. during configuration of the system.

Applicant further argues that "a person of ordinary skill in the art, considering the art of Sicola and Mikashayeki, would not arrive at the present invention. Mikashayeki explicitly teaches a method to enable ‘failover policies for a cluster environment having more than two nodes in which the applications running on a failed node may be dynamically allocated.' A combination of Sicola and Mikashayeki would, determine failover on the ‘weight of every one of the surviving nodes.’ Conversely, the current invention is concerned with ‘replication paths’ not the weight of a node. For, if 'another Mobility Agent cannot be found on the same replication path the replication will fail'. However, if 'there is more than one failover agent in the environment, the order of failover is decided during configuration of the architecture.’" In response to this argument, Examiner notes that, as stated above, Mikashayeki et al. teaches “known failover policies” that include methods that are not determined as to the current weight of the nodes (see 1:58-2:16). In addition to that, the other portion of Mikashayeki et al. cited by the Examiner (see 6:25-38) does not determine failover "on the weight of every one of the surviving nodes”, but rather only to “the successive node or the previous node”. Thus, Mikashayeki et al. is still concerned with ‘replication paths’, as claimed by the Applicant. In addition to that, the setup of determining successive and previous nodes is predetermined and fixed, unlike later setups of Mikashayeki et al. in which nodes may be added to the system.

Applicants further assert that “another difference between Mikashayeki and the current invention is that Mikashayeki includes ‘assigning the applications running on the failed node to the failover node,' whereas the current invention 'will re-issue the command to another Mobility Agent on the same replication path". Applicant goes on to state "this means that Mikashayeki 'assigns applications', where in the current invention the 'software agent' 'being remote from the primary software agent,’ is already present and running. Applicants assert that Sicola does not disclose ‘failover software agents being remote from the primary software agent … each failover software agent residing on a host.' Rather, Sicola states that ‘no special host software is required to implement the above features because all replication functionality is totally self contained within each array controller". In response to this argument, it is noted that the rejection is based on the teachings of Sicola et al. in view of Mikashayeki et al., and that, as stated above, Sicola et al. teaches failover software agents being remote from the primary software agent … each failover software agent residing on a host.

Applicant then asserts “that one skilled in the art combining Mikashayeki with Sicola would not arrive at the current invention. First, Sicola, by claiming it is not necessary, teaches away from having ‘special host software”. This argument has already been addressed above .

"Second, Mikashayeki would not have need for ‘failover software agents being remote from the primary software agent' as it is a method to ‘assigning applications running on the failed node to the failover node.” Applicants assert that ‘assigning applications running on the failed node to the failover node’ would remove the need for one skilled in the art to place ‘failover software agents being remote from the primary software agent’”. It is noted that Sicola et al. teaches failover software agents being remote from the primary software agent, as stated above.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHARLES D. ADAMS whose telephone number is (571)272-3938. The examiner can normally be reached on 8:30 AM - 5:00 PM, M - F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Charles Rones can be reached on (571) 272-4085. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/C. D. A./
Examiner, Art Unit 2164

/Sathyanarayan Pannala/
Primary Examiner, Art Unit 2164